PROTOTYPE-ROUTED FLOW MATCHING FOR INCOMPLETE MULTI-VIEW CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Incomplete multi-view clustering (IMVC) aims to learn cluster-discriminative representations from partially observed views. Completion-based methods estimate missing-view representations so that observed and completed views can be used together for clustering. However, direct prediction and prototype-based reconstruction typically produce a single estimate for each missing view. When the observed views cannot uniquely determine the missing view, a single estimate may weaken cluster separation. Generative completion can instead model a conditional distribution over missing-view representations, but iterative denoising makes diffusion-based completion costly during training. We propose Prototype-Routed Flow Matching (PRFM), which formulates missing-view completion as clustering-aligned conditional transport in a shared latent space. PRFM uses dynamically updated online prototypes as routing signals during flow matching, transforming missing-view completion from static point prediction into a prototype-routed transport process. At each generation step, the assigned prototype provides a cluster-level direction, while a residual velocity captures sample- and view-specific variation. The same prototypes also prioritize observable candidates for cluster-consistent soft supervision and optimize completed representations in the prototype space. Completed latents are produced through few-step ODE inference. Experiments on multiple IMVC benchmarks show that PRFM achieves competitive clustering performance, remains robust under high missing rates, and offers favorable training and inference efficiency compared with diffusion-based completion. The code will be released upon acceptance.

1 INTRODUCTION

Multi-view clustering aims to discover semantic groups by exploiting complementary information across modalities or feature views (Zhou et al., 2026; Yuan et al., 2025). In practice, samples may have only a subset of views available (Wang et al., 2026a; Gu et al., 2026; Peng et al., 2026; Wang et al., 2026b). Samples from different clusters may look similar in the observed views, even when the missing views would distinguish them. The challenge in incomplete multi-view clustering (IMVC) is therefore to learn cluster-discriminative representations from incomplete observations.

Existing completion methods broadly follow two approaches: (i) endpoint-based methods directly predict or reconstruct a missing-view representation without modeling its generation path (Chao et al., 2024; Tang & Liu, 2022; Yuan et al., 2026), and (ii) generative methods model a process for producing it. Sample-wise predictors directly map observed views to a missing-view estimate (Lin et al., 2021; 2023), as illustrated in the upper branch of Figure 1(a). This is efficient, but fitting one estimate to several plausible missing-view patterns may average their semantics, placing the result between cluster regions and weakening cluster discrimination. Prototype-based completion introduces cluster-level structure into reconstruction (Li et al., 2023; Yuan et al., 2024). These methods form cluster-aware point estimates using sample-prototype relations or combinations of target-view prototypes. When several prototypes receive similar weights, their combination may still blur cluster distinctions in the completed representation, as illustrated in the lower branch of Figure 1(a). Because the representation is reconstructed in one step, prototypes influence the final estimate without guiding a generation trajectory. This motivates modeling conditional variation beyond a point estimate while retaining the cluster structure supplied by prototypes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

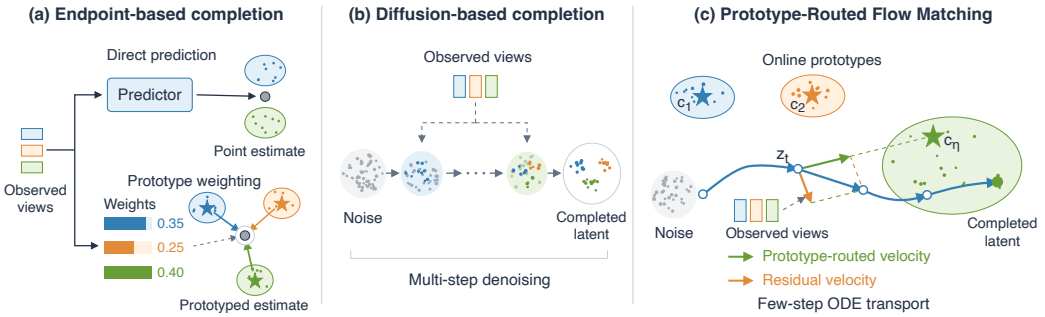


Figure 1: Schematic comparison of missing-view completion mechanisms. (a) Endpoint-based completion produces a point estimate through direct prediction or prototype-based reconstruction. Ambiguous predictions or similar weights across prototypes may blur cluster distinctions. (b) Diffusion-based completion generates missing-view latents from noise through conditional denoising, but requires repeated sampling steps. (c) PRFM performs few-step conditional transport by combining a prototype-routed velocity, constructed from online prototypes for cluster-level guidance, with a residual velocity that captures sample- and view-specific details.

Generative completion provides a way to model this uncertainty through a conditional distribution over missing-view representations. Diffusion-based methods generate missing-view representations through iterative denoising (Wen et al., 2024; Zhang et al., 2025; Guo et al., 2026), as shown in Figure 1(b), but repeated sampling can be costly when completion is invoked throughout training. Flow matching (Lipman et al., 2023) offers an alternative: it learns a velocity field that transports noise to target latents, and we use a few ODE steps to generate missing-view representations. Simply providing a prototype as a conditioning input, however, does not explicitly direct the velocity toward the assigned prototype; that behavior must be learned by the network. This motivates incorporating prototype guidance into the velocity field itself while retaining flexibility for sample- and view-specific variation.

We propose Prototype-Routed Flow Matching (PRFM), a framework for cluster-aligned conditional transport in a shared latent space (Figure 1(c)). PRFM introduces online clustering prototypes into the missing-view completion process. Each prototype is an evolving cluster-level semantic center learned from the current representations. During flow matching, the assigned prototype is not used merely as a condition or an endpoint constraint; instead, it directly provides a prototype-routed velocity that guides the current flow state toward the corresponding cluster region. A learned residual velocity complements this route by modeling the remaining target-view correction and preserving sample- and view-specific variation. As generation proceeds, the prototype contribution gradually decreases, so the process shifts from cluster-level routing to residual refinement. The same prototypes are also used beyond generation. When direct paired observations are unavailable, PRFM selects observable samples assigned to the same prototype to build soft supervision targets. The observed and generated latents are then fused and optimized in the same prototype space.

Our contributions are summarized as follows:

- We formulate missing-view completion as clustering-aligned conditional transport in a shared latent space. Specifically, we use dynamically updated prototypes to explicitly construct the velocity field: a state-dependent prototype-routed velocity provides a cluster-level direction toward the assigned prototype, while a learned residual velocity models sample- and view-specific corrections.
- We design prototype-prioritized soft pairing, which constructs soft supervision targets from observable candidates with the same inferred prototype assignment, alleviating the lack of direct paired observations under high missing rates.
- We evaluate PRFM on four IMVC benchmarks, showing competitive clustering performance and robustness under high missing rates. Ablations support the proposed components, and controlled comparisons with diffusion-based completion demonstrate favorable training-step and inference latency.

2 Related Work

Incomplete Multi-View Clustering learns cluster-discriminative representations from partially observed multi-view data. While imputation-free methods aggregate only available views (Dai et al., 2025; Xu et al., 2024), completion-based methods recover missing views or latent representations to exploit cross-view complementarity. Sample-wise point completion methods predict missing representations from observed views through cross-view prediction or neighbor relations (Tang & Liu, 2022; Chao et al., 2024), but they often produce a single estimate for each missing view. Prototype-based completion further introduces cluster-level information through sample-prototype relations or prototype combinations (Li et al., 2023; Yuan et al., 2024). Although more cluster-aware, these methods mainly use prototypes as static templates or target anchors to construct the final completed representation, rather than to explicitly shape a continuous generation process. Diffusion-based completion can model richer conditional distributions (Wen et al., 2024; Zhang et al., 2025; Guo et al., 2026), but its iterative denoising process is costly when completion is repeatedly performed during training. PRFM instead uses dynamically updated prototypes to explicitly shape the velocity field, aligning missing-view completion with cluster-discriminative representation learning.

Flow Matching learns a transport process from a simple noise distribution to a target distribution by directly regressing a velocity field (Lipman et al., 2023; Liu et al., 2023; Tong et al., 2024; Albergo et al., 2025; Kornilov et al., 2024). Compared with diffusion models, its direct velocity regression and few-step ODE inference make it suitable for completion-based IMVC, where missing-view generation is invoked throughout training. However, standard conditional flow matching mainly learns transport toward the conditional target distribution, with no explicit cluster guidance in the velocity field. PRFM adapts flow matching to IMVC by using dynamically updated prototypes to shape this field: the prototype-routed component provides cluster-level direction, while the residual velocity captures sample- and view-specific details.

3 Method

3.1 Problem Formulation

Let $\mathcal{X} = \{(x_i^1, \ldots, x_i^V)\}_{i=1}^N$ denote a multi-view dataset with N samples and V views, where $x_i^v \in \mathbb{R}^{d_v}$ represents the feature of sample i in view v, and the feature dimension d_v may vary across views. In the incomplete setting, a binary mask $M \in \{0,1\}^{N\times V}$ indicates view availability, where $m_i^v = 1$ if x_i^v is observed and $m_i^v = 0$ otherwise. IMVC aims to partition all samples into K disjoint clusters using only the available views, without label supervision.

3.2 Shared Semantic Encoder

As shown in Figure 2, we first pretrain the encoders to establish a shared latent space. For each observed view, a view-specific encoder E_{ψ_v} followed by a shared Transformer encoder P_ϕ produces

$$z_i^v = P_\phi(E_{\psi_v}(x_i^v)), \quad m_i^v = 1, \tag{1}$$

where $z_i^v \in \mathbb{R}^{d_z}$ is the latent representation of view v. Here E_{ψ_v} handles the heterogeneous input dimension of each view, while the Transformer encoder P_ϕ is shared across all views to map them into a common semantic space. The observed view latents are then fused into a consensus representation $z_i = \sum_{v:m_i^v=1} w_i^v z_i^v / \sum_{v:m_i^v=1} w_i^v$, where w_i^v is the predicted confidence of view v. Paired observed views of the same sample are aligned by a symmetric InfoNCE loss (van den Oord et al., 2018), denoted as $\mathcal{L}_{\text{ctr}}$, which is used for encoder pretraining and retained as an alignment regularizer. Implementation details and the full contrastive objective are provided in Appendix A.

3.3 Online Prototype Learning

After encoder pretraining, we apply K-means++ (Arthur & Vassilvitskii, 2007) to the observed-view consensus representations $\{z_i\}_{i=1}^N$ and normalize the resulting centers to initialize K prototypes $\{c_k \in \mathbb{R}^{d_z}\}_{k=1}^K$. During subsequent training, the soft assignment of sample i is computed as

$$q_{ik} = \frac{\exp(\text{sim}(z_i, c_k)/\tau_{\text{asg}})}{\sum_{k'=1}^K \exp(\text{sim}(z_i, c_{k'})/\tau_{\text{asg}})}, \tag{2}$$

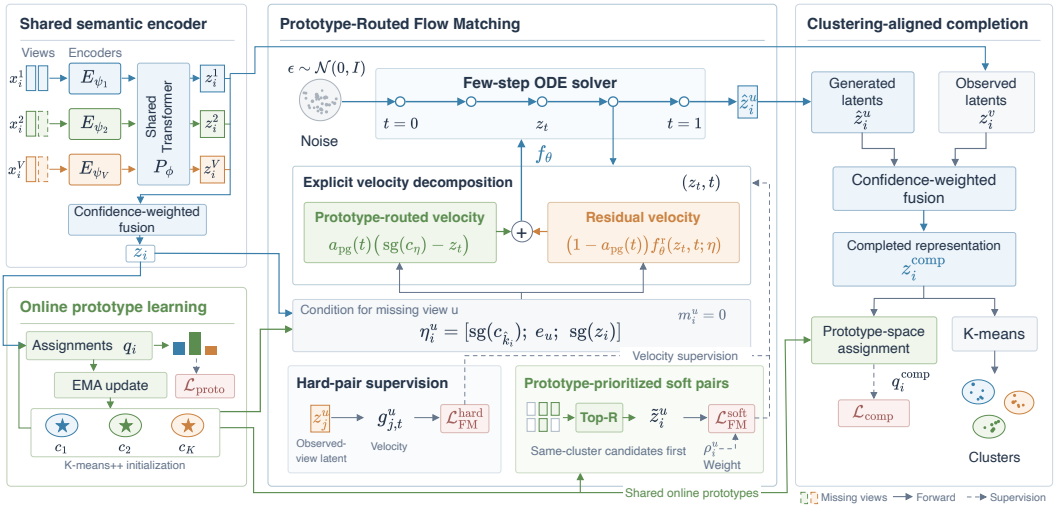


Figure 2: Overview of PRFM. Observed views are encoded and fused into a consensus representation, which is used to learn online prototypes. For a missing view, the assigned prototype explicitly shapes the flow-matching velocity field through a prototype-routed component, while the residual velocity captures sample- and view-specific details. Hard-pair and prototype-prioritized soft-pair supervision train the generator in the latent space. Generated and observed latent views are then fused into a completed representation and optimized in the same prototype space. Solid arrows denote forward computation, and dashed arrows denote supervision or candidate selection.

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity and τ_{asg} is an annealed assignment temperature. The hard assignment $\hat{k}_i = \arg\max_k q_{ik}$ determines the prototype $c_{\hat{k}_i}$ used by the completion module. To optimize assignments, we compute the mini-batch assignment mass $\nu_k = \sum_{i\in\mathcal{B}} q_{ik}$ and construct a frequency-normalized sharpened target:

$$\tilde{q}_{ik} = \frac{q_{ik}^2/(\nu_k+\varepsilon)}{\sum_{k'=1}^{K} q_{ik'}^2/(\nu_{k'}+\varepsilon)}, \tag{3}$$

where $\varepsilon > 0$ is a smoothing constant. Let $\bar{q}_k = \nu_k/|\mathcal{B}|$ denote the average usage of prototype k. The prototype objective is

$$\mathcal{L}_{\text{proto}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \mathrm{KL}(\mathrm{sg}(\tilde{q}_i)\,\|\,q_i) + \sum_{k=1}^{K} \bar{q}_k \log \bar{q}_k, \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The first term promotes confident assignments, while the second encourages balanced prototype usage. The prototype bank is updated online by EMA, with details provided in Appendix A.

3.4 Prototype-Routed Flow Matching

We formulate missing-view completion as conditional latent transport from Gaussian noise to a target-view representation. We use a straight-line path for flow matching (Lipman et al., 2023). Given a supervised endpoint $z_{\text{end}} \in \mathbb{R}^{d_z}$, we draw $\epsilon \sim \mathcal{N}(0, I_{d_z})$ and $t \sim \mathrm{Unif}(0,1)$, and form $z_t = (1-t)\epsilon + t z_{\text{end}}$. Its target velocity is $g_t = z_{\text{end}} - \epsilon$. Flow matching trains $f_\theta(z_t, t; \eta)$ to predict g_t by squared-error regression, where η is the completion condition defined below. At inference, the learned field defines an ODE that transports noise to the completed latent. The endpoint z_{end} is an observed target-view latent for hard pairs or a soft endpoint constructed from observed candidates for missing views. Let $u \in \{1,\dots,V\}$ denote the view to be completed. For a missing pair (i,u), the condition is

$$\eta_i^u = \left[\mathrm{sg}(c_{\hat{k}_i}); e_u; \mathrm{sg}(z_i)\right] \in \mathbb{R}^{3d_z}, \tag{5}$$

where $e_u \in \mathbb{R}^{d_z}$ is a learnable vector associated with view u. The three entries specify the assigned prototype, the target view, and the available-view consensus of the current sample, respectively.

Prototype-routed velocity field. The core of PRFM is to make the prototype act on the transport process itself. In prototype-based point completion, prototypes mainly constrain the completed endpoint. In standard conditional flow matching, prototypes can be appended as conditions, but they only influence the velocity indirectly through the network output. PRFM instead parameterizes the velocity field with an explicit prototype-routed component. For a generic condition η, let c_η denote the prototype carried by η. We decompose the velocity as

$$\begin{aligned} f_\theta(z_t, t; \eta) &= \underbrace{a_{\mathrm{pg}}(t)\,(\mathrm{sg}(c_\eta) - z_t)}_{\text{prototype-routed velocity}} + \underbrace{(1 - a_{\mathrm{pg}}(t))\, f_\theta^{\mathrm{r}}(z_t, t; \eta)}_{\text{residual velocity}}, \\ a_{\mathrm{pg}}(t) &= \lambda_{\mathrm{pg}}(1-t)^{p_{\mathrm{pg}}}, \end{aligned} \tag{6}$$

where f_θ^{r} is the residual network, $\lambda_{\mathrm{pg}} \in [0, 1)$ controls the initial routing strength, and $p_{\mathrm{pg}} > 0$ controls its decay. The prototype-routed velocity gives the flow state an explicit cluster-level direction by moving z_t toward the prototype carried in the condition. This prevents the trajectory from being determined only by the residual network and makes the generation process aware of the assigned cluster from the beginning. The residual velocity then learns the complementary correction required for the target view, preserving instance- and view-specific details beyond the cluster-level prototype. Since $a_{\mathrm{pg}}(t)$ decreases with t, PRFM emphasizes prototype routing near the noisy initial state and gradually shifts to residual refinement near the endpoint.

Hard-pair supervision. When view u of sample j is observed and at least one other view is available, we use z_j^u as a supervised target for missing-view generation. We compute the conditioning consensus from the other observed views and use it to select one of the shared online prototypes. The hard-pair condition $\eta_{j,u}^{\mathrm{hard}}$ contains this consensus, the selected prototype, and the target-view embedding e_u, which identifies the view to generate. The observed latent $\mathrm{sg}(z_j^u)$ is not included in the condition and serves as the endpoint of the training path. With $\epsilon \sim \mathcal{N}(0, I_{d_z})$ and $t \sim \mathrm{Unif}(0, 1)$, the path and target velocity are $z_{j,t}^u = (1-t)\epsilon + t\,\mathrm{sg}(z_j^u)$ and $g_{j,t}^u = \mathrm{sg}(z_j^u) - \epsilon$. These valid hard pairs form $\mathcal{H} = \{(j, u) \mid m_j^u = 1,\ \sum_{v \neq u} m_j^v > 0\}$. The hard flow-matching loss is

$$\mathcal{L}_{\mathrm{FM}}^{\mathrm{hard}} = \mathbb{E}_{\substack{(j,u)\in\mathcal{H} \\ t,\epsilon}} \left[\left\| f_\theta\big(z_{j,t}^u, t; \eta_{j,u}^{\mathrm{hard}}\big) - g_{j,t}^u \right\|_2^2 \right]. \tag{7}$$

Prototype-prioritized soft-pair supervision. Hard pairs are unavailable when the target view is missing. For such pairs, PRFM uses prototypes to construct a soft endpoint from observable candidates. For a missing pair (i, u), we first collect mini-batch candidates whose view u is observed and whose assigned cluster matches that of sample i:

$$\mathcal{A}_i^u = \{j \in \mathcal{B} \mid m_j^u = 1,\ \hat{k}_j = \hat{k}_i\}.$$

If this set is empty, we fall back to all samples with view u observed:

$$\widetilde{\mathcal{A}}_i^u = \begin{cases} \mathcal{A}_i^u, & \text{if } \mathcal{A}_i^u \neq \emptyset, \\ \{j \in \mathcal{B} \mid m_j^u = 1\}, & \text{otherwise.} \end{cases} \tag{8}$$

Each candidate is scored by $s_{ij}^u = \mathrm{sim}(z_j^u, c_{\hat{k}_i})$. Let $\mathcal{A}_{i,R}^u$ contain the top-R candidates, or all candidates if fewer than R are available. The prototype- prioritized soft endpoint is

$$\alpha_{ij}^u = \frac{\exp(s_{ij}^u/\tau_p)}{\sum_{j' \in \mathcal{A}_{i,R}^u} \exp(s_{ij'}^u/\tau_p)}, \qquad \widetilde{z}_i^u = \mathrm{sg}\left(\sum_{j \in \mathcal{A}_{i,R}^u} \alpha_{ij}^u z_j^u \right). \tag{9}$$

where $\tau_p > 0$ controls the softness of aggregation. As analyzed in Appendix D.2, the temperature controls whether the soft endpoint favors the closest candidates or averages local candidates. Let $\widetilde{z}_{i,t}^u = (1-t)\epsilon + t\widetilde{z}_i^u$ and $\widetilde{g}_{i,t}^u = \widetilde{z}_i^u - \epsilon$. We weight each soft pair by a confidence $\rho_i^u \in [0, 1]$ computed from the assignment confidence of the anchor and selected candidates. Let $\mathcal{S} = \{(i, u) \mid m_i^u = 0,\ \widetilde{\mathcal{A}}_i^u \neq \emptyset\}$ denote valid missing pairs for soft supervision. The soft flow-matching loss is

$$\mathcal{L}_{\mathrm{FM}}^{\mathrm{soft}} = \mathbb{E}_{\substack{(i,u)\in\mathcal{S} \\ t,\epsilon}} \left[\rho_i^u \left\| f_\theta\big(\widetilde{z}_{i,t}^u, t; \eta_i^u\big) - \widetilde{g}_{i,t}^u \right\|_2^2 \right]. \tag{10}$$

The complete flow-matching objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathcal{L}_{\mathrm{FM}}^{\mathrm{hard}} + \lambda \mathcal{L}_{\mathrm{FM}}^{\mathrm{soft}}, \tag{11}$$

where $\lambda \geq 0$ balances hard- and soft-pair supervision.

Inductive bias and regression compatibility. The following proposition shows that the residual field can compensate for the prototype term, preserving the weighted velocity-regression optimum when the residual field is unrestricted.

Proposition 1 (Velocity decomposition under weighted regression) *Fix the latent representations, conditioning inputs, and prototype bank, and let c_η denote the assigned prototype carried by condition η. Let $X_t = (z_t, t, \eta)$, $a_t = a_{\mathrm{pg}}(t) \in [0, 1)$, $b_t = \mathrm{sg}(c_\eta) - z_t$, and let g_t be a sampled target velocity drawn from hard- or soft-pair supervision. Let $\xi \geq 0$ denote the supervision weight, including pair-type sampling normalization as specified in Appendix D.1. With ξ fixed with respect to the velocity parameters, assume $\mathbb{E}[\xi] < \infty$, $\mathbb{E}[\xi \|g_t\|^2] < \infty$, and $0 < \mathbb{E}[\xi \mid X_t] < \infty$ on the states considered. Over unrestricted measurable residual fields, the minimizer of $\mathbb{E}[\xi \|a_t b_t + (1 - a_t) f^{\mathrm{r}}(X_t) - g_t\|^2]$ satisfies, almost surely on these states,*

$$\begin{aligned} f^{\mathrm{r}\star}(X_t) &= \frac{u_\xi(X_t) - a_t b_t}{1 - a_t}, \\ f^\star(X_t) := a_t b_t + (1 - a_t) f^{\mathrm{r}\star}(X_t) &= u_\xi(X_t) = \frac{\mathbb{E}[\xi g_t \mid X_t]}{\mathbb{E}[\xi \mid X_t]}. \end{aligned} \tag{12}$$

Thus, the residual can offset the prototype term at the weighted regression optimum. Prototype routing changes the velocity parameterization without changing the optimal total field in this unrestricted setting; its effect as a cluster-level inductive bias under finite-capacity training is evaluated empirically. The proof and geometric scope of prototype attraction are provided in Appendix D.1.

Inference. For each missing pair (i, u), we sample $\epsilon \sim \mathcal{N}(0, I_{d_z})$ and solve

$$\frac{\mathrm{d}\gamma_i^u(t)}{\mathrm{d}t} = f_\theta(\gamma_i^u(t), t; \eta_i^u), \quad \gamma_i^u(0) = \epsilon, \quad \widehat{z}_i^u = \gamma_i^u(1). \tag{13}$$

The generated latent $\widehat{z}_i^u$ is subsequently combined with the observed views for clustering-aligned completion.

3.5 Clustering-Aligned Completion

PRFM aggregates the observed and generated latent views into a completed consensus representation:

$$\bar{z}_i^v = \begin{cases} z_i^v, & m_i^v = 1, \\ \widehat{z}_i^v, & m_i^v = 0, \end{cases} \qquad z_i^{\mathrm{comp}} = \frac{\sum_{v=1}^V \omega_i^v \bar{z}_i^v}{\sum_{v=1}^V \omega_i^v}. \tag{14}$$

where $\bar{z}_i^v$ uses the observed latent if available and the generated latent otherwise. Observed-view weights use encoder confidence, while generated-view weights are adjusted by their agreement with the observed-view consensus and assigned prototype. The detailed weighting rule is provided in Appendix A. The completed representation is then assigned in the same prototype space:

$$q_{ik}^{\mathrm{comp}} = \frac{\exp(\mathrm{sim}(z_i^{\mathrm{comp}}, c_k)/\tau_{\mathrm{asg}})}{\sum_{k'=1}^K \exp(\mathrm{sim}(z_i^{\mathrm{comp}}, c_{k'})/\tau_{\mathrm{asg}})}. \tag{15}$$

Let $\widetilde{q}_i^{\mathrm{comp}}$ denote the sharpened target constructed from q_i^{comp} in the same way as the prototype target, and let $\widetilde{q}_i$ denote the sharpened target from observed-view assignments. The clustering-aligned completion objective is

$$\mathcal{L}_{\mathrm{comp}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[\mathrm{KL}(\mathrm{sg}(\widetilde{q}_i^{\mathrm{comp}}) \,\|\, q_i^{\mathrm{comp}}) + \mathrm{KL}(\mathrm{sg}(\widetilde{q}_i) \,\|\, q_i^{\mathrm{comp}}) \right], \tag{16}$$

where the first term sharpens the completed assignments, while the second aligns them with the cluster structure learned from the observed views. The summation is over all samples in the mini-batch; when a sample has no missing view, z_i^{comp} is simply the fused observed-view representation.

At evaluation, we apply K-means to the completed representations $\{z_i^{\mathrm{comp}}\}_{i=1}^N$ to obtain the final clustering results. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{ctr}} + \mathcal{L}_{\mathrm{proto}} + \mathcal{L}_{\mathrm{FM}} + \mathcal{L}_{\mathrm{comp}}. \tag{17}$$

4 EXPERIMENTS

We evaluate PRFM from both effectiveness and efficiency perspectives. Our study is guided by seven questions: **Q1:** Does PRFM outperform existing IMVC methods across datasets and missing rates (Sec. 4.2)? **Q2:** Are the proposed components crucial for the improvements (Sec. 4.3)? **Q3:** Does PRFM achieve lower cost than diffusion-based completion (Sec. 4.4)? **Q4:** Does PRFM produce more cluster-discriminative completed representations (Sec. 4.5)? **Q5:** How sensitive is PRFM to key hyper-parameters and Euler steps (Appendix E)? **Q6:** Does PRFM converge stably (Appendix F)? **Q7:** How does PRFM compare with representative completion methods in runtime (Appendix G)?

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate PRFM on four widely used datasets: MSRC-V1 (Winn et al., 2005), HandWritten (LeCun et al., 1989), ALOI-Deep3V (Geusebroek et al., 2005), and YouTubeFace10 (Wolf et al., 2011). Detailed descriptions of these datasets are provided in the appendix.

Baselines. We compare PRFM against seven state-of-the-art IMVC methods, including DCP (Lin et al., 2023), ProImp (Li et al., 2023), RPCIC (Yuan et al., 2024), BURG (Jin et al., 2025), FreeCSL (Dai et al., 2025), DCG (Zhang et al., 2025), and NCD (Guo et al., 2026).

Implementation Details. We implement PRFM in PyTorch (Paszke et al., 2019). Across all datasets, view-specific MLP encoders and a shared Transformer map observed views into a common latent space. We pretrain the encoders with contrastive alignment and then jointly optimize prototype learning, flow matching, and clustering-aligned completion. Full settings are provided in Appendix A.

4.2 PERFORMANCE COMPARISON (Q1)

In Table 1, we compare PRFM with seven state-of-the-art IMVC methods. PRFM achieves the best ACC in all settings and the best or second-best ARI in all settings. For example, its advantage is most pronounced on MSRC-V1. At a missing rate of 70%, PRFM improves ACC from the strongest baseline result of 61.43 to 79.05 and ARI from 42.58 to 59.78. On ALOI-Deep3V, HandWritten, and YouTubeFace10, PRFM also consistently obtains the best ACC and maintains stable ARI performance, indicating its adaptability to different data scales, category complexity, and missing rates. Although some baselines achieve slightly higher NMI in a few settings, PRFM provides more stable advantages in ACC and ARI, leading to stronger overall clustering quality. These results show that PRFM delivers robust clustering performance across diverse incomplete multi-view benchmarks, especially under high missing rates.

4.3 ABLATION STUDY (Q2)

Table 2 evaluates the contribution of each core component at a missing rate of 70%. We first isolate the completion mechanism: *Deterministic Completion* replaces flow matching with a direct MLP regressor, while *Standard FM* keeps flow matching but removes prototype conditioning and prototype-routed velocity. We then separate the two uses of prototypes in the transport field: *Prototype-Cond. FM* keeps prototypes as ordinary conditioning inputs but removes the explicit prototype-routed velocity term, while *w/o* Residual Velocity keeps only the prototype-routed velocity. Finally, *w/o* Obs. Rep., *w/o* Prototype-Prioritized Weighting, and *w/o* Soft Pair. remove the observed-view consensus representation, replace prototype-prioritized aggregation with uniform aggregation, and remove soft pairing, respectively.

Table 1: Clustering performance under different missing rates (MR). The best and second-best results are marked in **bold** and underline, respectively.

MR	Method	MSRC-V1			HandWritten			ALOI-Deep3V			YouTubeFace10		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
10%	DCP	21.90	15.38	0.47	91.40	85.13	81.96	33.31	77.83	27.52	67.16	75.85	51.02
	ProImp	60.48	53.45	42.46	72.10	72.64	61.71	88.30	**97.40**	88.69	50.33	55.80	31.52
	RPCIC	62.38	52.56	41.93	81.35	79.89	73.96	79.43	94.77	80.53	73.50	78.97	65.87
	BURG	66.19	59.40	50.54	82.30	84.22	77.37	67.37	88.60	57.48	73.90	78.15	62.80
	FreeCSL	77.62	71.26	63.68	81.70	86.68	79.35	87.02	97.15	87.89	74.05	**79.53**	68.87
	DCG	49.05	39.15	25.82	82.75	82.63	74.88	63.13	86.99	52.82	76.93	78.39	69.39
	NCD	74.76	67.97	59.99	93.60	87.35	86.59	92.55	97.02	91.01	81.16	78.68	70.64
	Ours	**83.81**	**71.66**	**66.75**	**94.90**	**89.16**	**89.06**	**92.94**	97.24	**91.36**	**81.34**	78.99	**71.76**
30%	DCP	20.48	13.47	0.41	85.15	75.04	70.80	25.69	72.63	19.99	64.01	70.57	38.36
	ProImp	61.90	50.98	39.18	82.65	78.93	73.59	89.61	**97.57**	89.78	49.10	49.88	30.29
	RPCIC	51.43	41.41	26.74	80.80	78.54	71.52	79.65	95.15	80.94	72.84	77.19	60.29
	BURG	65.71	59.72	50.19	66.45	72.19	57.57	59.24	80.32	34.95	65.37	70.73	53.34
	FreeCSL	73.81	69.09	61.06	84.80	**83.96**	78.44	85.30	96.67	86.00	72.63	**77.96**	67.53
	DCG	48.57	40.47	26.47	82.70	80.54	73.96	61.59	86.14	49.63	72.17	71.57	61.98
	NCD	78.57	68.53	57.26	90.35	83.69	80.06	92.26	96.24	89.02	80.36	74.57	67.29
	Ours	**82.86**	**71.35**	**65.36**	**91.10**	83.04	**81.39**	**92.49**	96.90	**90.48**	**80.51**	75.19	**70.04**
50%	DCP	21.43	14.16	0.28	79.85	75.21	59.49	24.59	68.47	17.21	64.45	72.41	43.37
	ProImp	71.43	57.27	47.42	66.30	66.77	52.84	87.96	**96.70**	87.57	44.45	50.66	28.00
	RPCIC	40.48	34.03	17.32	80.05	79.61	72.47	79.56	94.75	79.56	74.26	74.76	61.64
	BURG	63.33	60.43	50.61	61.45	66.95	48.17	59.31	78.81	21.51	64.27	66.70	48.27
	FreeCSL	66.67	57.51	48.79	79.75	80.96	73.60	83.18	93.23	79.88	65.89	71.43	58.60
	DCG	52.86	45.86	33.51	80.80	76.21	70.45	53.17	81.06	46.09	71.27	**76.66**	64.49
	NCD	40.95	30.55	17.88	88.32	**81.91**	74.75	91.34	96.16	88.49	74.61	74.23	66.99
	Ours	**79.05**	**66.41**	**59.36**	**88.65**	79.77	**76.87**	**91.53**	96.22	**88.89**	**75.09**	76.08	**68.67**
70%	DCP	22.86	16.71	1.00	71.70	72.22	52.98	32.93	76.33	22.60	63.03	70.54	42.82
	ProImp	49.05	41.97	27.09	77.25	68.69	61.37	86.23	**96.30**	85.91	49.36	52.77	34.16
	RPCIC	37.14	29.36	14.48	78.05	**77.63**	**70.14**	80.16	94.65	79.89	73.46	**77.47**	63.53
	BURG	61.43	51.89	42.58	61.00	63.94	45.18	61.64	82.55	46.59	55.77	59.33	38.75
	FreeCSL	59.05	49.47	37.98	77.55	73.12	67.90	58.99	84.01	54.26	64.32	70.64	57.77
	DCG	55.24	40.77	27.74	79.75	74.00	67.54	54.64	79.38	46.91	69.47	70.84	59.11
	NCD	52.86	38.50	27.37	82.95	77.27	68.23	88.52	87.79	86.99	77.49	72.39	64.20
	Ours	**79.05**	**64.83**	**59.78**	**83.70**	72.96	68.50	**91.85**	96.29	**89.19**	**78.72**	73.92	**66.14**

Table 2: Ablation study of PRFM on HandWritten and YouTubeFace10 at a missing rate of 70%, where *w/o* denotes that the corresponding component is removed.

Variant	HandWritten			YouTubeFace10		
	ACC	NMI	ARI	ACC	NMI	ARI
Full PRFM	**83.70**	**72.96**	**68.50**	**78.72**	**73.92**	**66.14**
Deterministic Completion	81.15	68.39	64.02	71.66	71.15	62.13
Standard FM	78.60	66.54	60.17	69.34	63.85	53.24
Prototype-Cond. FM	81.10	69.52	64.21	70.62	61.92	53.66
w/o Residual Velocity	82.30	70.96	66.08	78.15	70.32	63.73
w/o Obs. Rep.	80.75	68.52	64.09	70.69	67.63	59.15
w/o Prototype-Prioritized Weighting	80.60	69.10	63.62	77.19	69.73	62.12
w/o Soft Pair.	79.40	67.40	61.27	69.63	64.55	54.03

The full model performs best on both datasets, confirming that the proposed components are complementary. Deterministic Completion is consistently worse than Full PRFM, especially on YouTube-Face10 where ACC drops from 78.72 to 71.66, indicating that direct point prediction is insufficient for complex missing-view completion. More importantly, Standard FM also performs clearly worse than Full PRFM, showing that a generic conditional transport model alone does not fully address the clustering requirement of IMVC.

Prototype-related variants further isolate where the gain comes from. Prototype-Cond. FM keeps prototypes as conditioning inputs but remains below Full PRFM, showing that the improvement is not merely due to adding prototypes as extra conditions. *w/o* Residual Velocity also underperforms Full PRFM, indicating that the residual velocity is needed to complement the prototype-routed direction with sample- and view-specific details. For soft supervision, *w/o* Prototype-Prioritized Weighting is worse than Full PRFM, and removing soft pairing entirely causes a larger drop. This confirms that prototype-prioritized aggregation provides more useful supervision than uniformly averaging

same-cluster candidates. Finally, *w/o* Obs. Rep. lowers the results, confirming that sample-specific observed information remains necessary in addition to cluster-level prototype guidance.

4.4 Runtime Analysis (Q3)

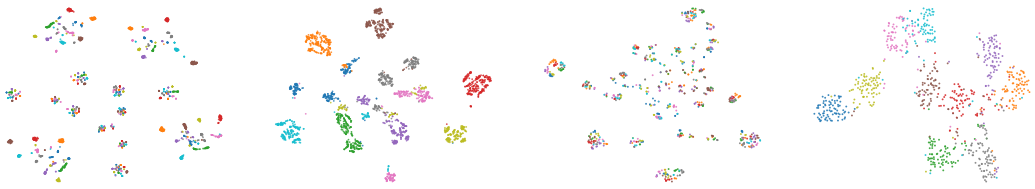
Table 3: Runtime comparison between PRFM and diffusion-based completion methods at a 70% missing rate. All methods are evaluated with the first two views, i.e., view pair $(1, 2)$, and the same batch size on each dataset.

Dataset	Method	Params (M)	Train Step (ms)	Inference Batch (ms)
HandWritten	DCG	15.80	200.17	151.79
HandWritten	NCD	24.64	**70.42**	28588.76
HandWritten	PRFM	**9.03**	85.76	**10.82**
YouTubeFace10	DCG	18.27	208.46	157.05
YouTubeFace10	NCD	25.87	94.34	56344.64
YouTubeFace10	PRFM	**9.65**	**87.44**	**10.86**

Table 3 compares PRFM with DCG and NCD under a controlled two-view runtime setting. The diffusion-based methods are measured using their official implementations, while PRFM uses the same view pair and batch size. PRFM uses fewer parameters and has competitive train-step latency, while achieving much lower inference latency because few-step ODE transport avoids the many iterative denoising steps required by diffusion-based completion.

4.5 t-SNE Visualization Analysis (Q4)

We further visualize the learned representations using t-SNE at a 70% missing rate. Figure 3 compares the original feature representations obtained from the observed views under the incomplete-view setting with the completed representations produced by PRFM on YouTubeFace10 and HandWritten. The original features are more scattered and contain substantial class mixing, especially when many views are missing. In contrast, the PRFM representations show more compact local groups and clearer class separation. This qualitative comparison suggests that PRFM produces completed representations with more cluster-discriminative structure.



(a) YTF original (b) YTF PRFM (c) HandWritten original (d) HandWritten PRFM

Figure 3: t-SNE visualization of original feature representations and PRFM completed representations at a 70% missing rate.

5 Conclusion

We presented PRFM for incomplete multi-view clustering, which formulates missing-view completion as clustering-aligned conditional transport in a shared latent space. PRFM uses dynamically updated online prototypes to shape the velocity field, while a residual velocity captures sample- and view-specific variation. The same prototypes are further used to construct soft supervision when direct paired observations are unavailable and to optimize completed representations in the prototype space. By coupling generation, supervision, and clustering, PRFM makes missing-view completion directly support cluster-discriminative representation learning. Experiments on multiple IMVC benchmarks show competitive clustering performance, robustness under high missing rates, and favorable training and inference efficiency compared with diffusion-based completion.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Guoqing Chao, Yi Jiang, and Dianhui Chu. Incomplete contrastive multi-view clustering with high-confidence guiding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11221–11229, 2024.

Yuzhuo Dai, Jiaqi Jin, Zhibin Dong, Siwei Wang, Xinwang Liu, En Zhu, Xihong Yang, Xinbiao Gan, and Yu Feng. Imputation-free and alignment-free: Incomplete multi-view clustering driven by consensus semantic learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5071–5081, 2025.

Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

Zhibin Gu, Zhenhao Zhong, Xi Zhang, and Bing Li. Constant degree matrix-driven incomplete multi-view clustering via connectivity-structure and embedding tensor learning. In *International Conference on Learning Representations*, 2026.

Qian Guo, Gaohui Zuo, Bingbing Jiang, Guangrui Fan, Zhihua Cui, Xinyan Liang, and Jianjian Ding. Incomplete multi-view clustering via neighborhood-conditioned diffusion. In *Proceedings of the Forty-Third International Conference on Machine Learning*, 2026.

Jiaqi Jin, Siwei Wang, Zhibin Dong, Xihong Yang, Xinwang Liu, En Zhu, and Kunlun He. Deep incomplete multi-view clustering with distribution dual-consistency recovery guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1016–1026, 2025.

Nikita Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *Advances in Neural Information Processing Systems*, 37:104180–104204, 2024.

Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Haobin Li, Yunfan Li, Mouxing Yang, Peng Hu, Dezhong Peng, and Xi Peng. Incomplete multi-view clustering via prototype-based imputation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3911–3919, 2023.

Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11174–11183, 2021.

Yijie Lin, Yuanbiao Gou, Xiaotian Liu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. Dual contrastive prediction for incomplete multi-view representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4447–4461, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Siyuan Peng, Shuzhao Xu, Jianye Yang, Yongyi Lu, and Feiping Nie. Pairwise constraint weighted imputation for incomplete multi-view clustering with high missing rate. *Information Fusion*, pp. 104599, 2026.

Huayi Tang and Yong Liu. Deep safe incomplete multi-view clustering: Theorem and algorithm. In *International conference on machine learning*, pp. 21090–21110. PMLR, 2022.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Qianqian Wang, Xurui Liao, Wei Feng, and Quanxue Gao. Debiased and denoised representation learning for incomplete multi-view clustering. In *International Conference on Learning Representations*, volume 2026, pp. 73338–73350, 2026a.

Qianqian Wang, Haiming Xu, Wei Feng, and Quanxue Gao. Adversarial fair incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10011–10019, 2026b.

Jie Wen, Shijie Deng, Waikeung Wong, Guoqing Chao, Chao Huang, Lunke Fei, and Yong Xu. Diffusion-based missing-view generation with the application on incomplete multi-view clustering. In *Forty-first international conference on machine learning*, volume 235, pp. 52762–52778. PMLR, 2024.

John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1800–1807, 2005.

Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 529–534, 2011.

Gehui Xu, Jie Wen, Chengliang Liu, Bing Hu, Yicheng Liu, Lunke Fei, and Wei Wang. Deep variational incomplete multi-view clustering: Exploring shared clustering structures. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16147–16155, 2024.

Honglin Yuan, Shiyun Lai, Xingfeng Li, Jian Dai, Yuan Sun, and Zhenwen Ren. Robust prototype completion for incomplete multi-view clustering. In *Proceedings of the ACM International Conference on Multimedia*, pp. 10402–10411, 2024.

Honglin Yuan, Xingfeng Li, Jian Dai, Xiaojian You, Yuan Sun, and Zhenwen Ren. Deep streaming view clustering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73526–73540. PMLR, 2025.

Honglin Yuan, Yuan Sun, Xingfeng Li, Xi Wu, Xiaojian You, and Zhenwen Ren. Adaptive hardness-driven dictionary distillation for incomplete streaming view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yuanyang Zhang, Yijie Lin, Weiqing Yan, Li Yao, Xinhang Wan, Guangyuan Li, Chao Zhang, Guanzhou Ke, and Jie Xu. Incomplete multi-view clustering via diffusion contrastive generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22650–22658, 2025.

Haochen Zhou, Guofeng Ding, Mouxing Yang, Peng Hu, Yijie Lin, and Xi Peng. Uncover underlying correspondence for robust multi-view clustering. In *International Conference on Learning Representations*, 2026.

APPENDIX

A IMPLEMENTATION DETAILS

Implementation of PRFM. We implement PRFM in PyTorch (Paszke et al., 2019). Each view is processed by a view-specific two-layer MLP encoder with hidden dimension 512, which maps the raw feature into $r_{\text{enc}} = 4$ latent vectors $U_i^v \in \mathbb{R}^{r_{\text{enc}} \times d_z}$. A shared Transformer encoder then prepends a learnable summary vector and outputs the view-level representation $z_i^v \in \mathbb{R}^{d_z}$. The Transformer has 4 layers, 4 attention heads, and dropout 0.1, with latent dimension $d_z = 256$. The flow is solved with four Euler updates by default. For online prototype learning, the number of prototypes K is set to the number of classes, the assignment temperature is annealed from 1.0 to 0.1, and prototypes are updated by exponential moving average with momentum 0.999.

Training Setup. We optimize PRFM with AdamW (Loshchilov & Hutter, 2019) and a cosine annealing schedule. The learning rates are 1×10^{-4} for the encoders and 3×10^{-4} for the flow generator, with weight decay 1×10^{-4}, gradient clipping 1.0, and minimum learning rate 1×10^{-6}. Training starts with contrastive pretraining of the encoders, followed by joint optimization of prototype-routed completion, clustering, and contrastive alignment. The batch size is 256 for MSRC-V1, HandWritten, and ALOI-Deep3V, and 512 for YouTubeFace10. We use $\lambda_{\text{pg}} = 0.5$, $p_{\text{pg}} = 1.0$, $\lambda = 0.05$, $\kappa_0 = 0.05$, and $p_{\text{conf}} = 2.0$ as the default hyper-parameter settings.

Datasets. We evaluate PRFM on four widely used multi-view benchmark datasets:

- **MSRC-V1** (Winn et al., 2005) consists of 210 images from seven semantic categories, including trees, buildings, airplanes, cows, faces, cars, and bicycles. Following common practice, we use GIST and HOG features as two views.
- **HandWritten** (LeCun et al., 1989) contains 2,000 handwritten digit images from 10 classes, corresponding to digits 0–9. Each sample is represented by six different feature views.
- **ALOI-Deep3V** (Geusebroek et al., 2005) contains 10,800 images from 100 object categories. We use three deep feature views extracted by ResNet50, VGG16, and Inception-v3 with pretrained weights.
- **YouTubeFace10** (Wolf et al., 2011) is a facial video dataset containing 38,654 samples from 10 subjects. We use LBP, HOG, GIST, and Gabor descriptors as four feature views for each sample.

Following Lin et al. (2023), we simulate incomplete multi-view data by randomly selecting instances according to the specified missing rate and masking between 1 and $V - 1$ views for each selected instance, while ensuring that at least one view remains observed.

Baselines. We compare PRFM with seven representative incomplete multi-view clustering baselines:

- **DCP** (Lin et al., 2023) learns incomplete multi-view representations through dual contrastive prediction.
- **ProImp** (Li et al., 2023) performs prototype-based imputation to recover missing-view representations.
- **RPCIC** (Yuan et al., 2024) improves the robustness of prototype completion under missing views.
- **BURG** (Jin et al., 2025) introduces distribution dual-consistency guidance for robust recovery.
- **FreeCSL** (Dai et al., 2025) learns consensus semantic representations without explicit imputation or alignment.
- **DCG** (Zhang et al., 2025) adopts diffusion contrastive generation for missing-view recovery.
- **NCD** (Guo et al., 2026) introduces neighborhood-conditioned diffusion that exploits stable cross-view neighborhood structures to guide robust missing-view latent recovery.

For all baselines, we use the authors' official implementations and follow the recommended hyper-parameter settings in the original papers. We evaluate all methods on all datasets under four missing rates $\{10\%, 30\%, 50\%, 70\%\}$. To reduce randomness, we repeat each experiment five times with different random seeds and report the mean performance.

Three widely-used metrics including Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI) are used for evaluation. A higher value of these metrics indicates a better clustering performance.

Prototype Target and Update Details. All assignment statistics are computed within the current mini-batch. For prototype k, we compute its assignment mass and assignment-weighted centroid as

$$\nu_k = \sum_{i \in \mathcal{B}} q_{ik}, \qquad \mu_k = \frac{\sum_{i \in \mathcal{B}} q_{ik} z_i}{\nu_k}. \tag{18}$$

Here $\zeta > 0$ is a small threshold, and μ_k is used only when $\nu_k > \zeta$. The prototypes are updated by exponential moving average:

$$c_k \leftarrow \begin{cases} \operatorname{norm}(\beta c_k + (1-\beta)\operatorname{norm}(\mu_k)), & \nu_k > \zeta, \\ c_k, & \text{otherwise}, \end{cases} \tag{19}$$

where $\operatorname{norm}(\cdot)$ denotes ℓ_2 normalization, β is the EMA momentum, and prototypes with negligible assignment mass are kept unchanged. The EMA update is performed without gradient propagation.

The sharpened assignment target and prototype objective are defined in Eqs. 3 and 4. In all experiments, the EMA momentum is set to $\beta = 0.999$.

Confidence-Weighted Completion. For observed views, the completion weight ω_i^v is the encoder confidence w_i^v. For generated views, we scale the mean observed-view confidence by an agreement score:

$$\omega_i^v = \gamma_g \bar{w}_i \left[\kappa_0 + (1-\kappa_0) \left(\frac{\operatorname{sim}(\widehat{z}_i^v, z_i) + \operatorname{sim}(\widehat{z}_i^v, c_{\hat{k}_i}) + 2}{4} \right)^{p_{\text{conf}}} \right], \quad m_i^v = 0, \tag{20}$$

where $\bar{w}_i$ is the average confidence over observed views. We use $\gamma_g = 0.25$, $\kappa_0 = 0.05$, and $p_{\text{conf}} = 2.0$ by default. Prototype-prioritized soft pairing is used during training whenever enabled.

Contrastive Alignment. To align the semantic representations of paired observed views, we employ the InfoNCE objective (van den Oord et al., 2018). We first define the normalized contrastive projection

$$h_i^v = \frac{H_{\text{ctr}}(z_i^v)}{\|H_{\text{ctr}}(z_i^v)\|_2}. \tag{21}$$

For each view pair $v < v'$, let $\mathcal{O}_{vv'} = \{i \mid m_i^v = m_i^{v'} = 1\}$ and $N_{vv'} = |\mathcal{O}_{vv'}|$. Over the valid pair set $\Omega = \{(v, v') \mid v < v',\ N_{vv'} \geq 2\}$, the symmetric contrastive loss is defined as

$$\mathcal{L}_{\text{ctr}} = \frac{1}{2|\Omega|} \sum_{(v,v') \in \Omega} \left(\ell_{v \to v'} + \ell_{v' \to v} \right), \tag{22}$$

where

$$\ell_{v \to v'} = -\frac{1}{N_{vv'}} \sum_{i \in \mathcal{O}_{vv'}} \log \frac{\exp\Big(\operatorname{sim}(h_i^v, h_i^{v'})/\tau_{\text{con}}\Big)}{\sum_{j \in \mathcal{O}_{vv'}} \exp\big(\operatorname{sim}(h_i^v, h_j^{v'})/\tau_{\text{con}}\big)}. \tag{23}$$

Here, $\operatorname{sim}(\cdot,\cdot)$ denotes cosine similarity, and τ_{con} is the contrastive temperature.

B Computational Complexity Analysis

Let B be the mini-batch size, V the number of views, K the number of prototypes, d_z the latent dimension, R the candidate size, and S the number of Euler steps. Denote by C_E and C_F the cost of encoding one view and evaluating the flow network once, respectively. For each mini-batch, view encoding costs $O(BVC_E)$, prototype assignment and update cost $O(BKd_z)$, supervised velocity

regression costs $O(BVC_F)$, and prototype-prioritized soft pairing costs $O(B^2Vd_z + BVRd_z)$. In addition, the clustering-aligned completion loss requires generating missing-view latents by few-step ODE sampling, which costs at most $O(BVSC_F)$ in the worst case. Since R and S are small constants in our implementation, the dominant training complexity is

$$O\big(BVC_E + BVC_F + BVSC_F + B^2Vd_z + BKd_z\big)\,. \tag{24}$$

At inference, PRFM performs at most S Euler steps for each missing view, giving the worst-case complexity

$$O(BVC_E + BVSC_F + BKd_z)\,. \tag{25}$$

With a small candidate size and only $S = 4$ Euler steps by default, PRFM keeps the completion overhead modest compared with diffusion-based completion methods that require many denoising steps.

C Algorithm Procedure of PRFM

The overall training procedure of PRFM is summarized in Algorithm 1.

Algorithm 1 Training Procedure of PRFM

Require: Incomplete multi-view data $\{x_i^v\}$, observation mask M, number of clusters K
Ensure: Final cluster assignments.
Pretrain the view encoders and shared Transformer with contrastive alignment (Eq. 22).
Encode observed views, form consensus representations (Eq. 1), and initialize prototypes by K-means++.
for each training epoch **do**
  for each mini-batch $\mathcal{B}$ **do**
    Encode available views and compute consensus representations z_i.
    Compute prototype assignments q_i and assigned prototypes $c_{\hat{k}_i}$ (Eq. 2).
    Prototype-routed generation: for each missing pair (i, u), build condition η_i^u and decompose the velocity into the prototype-routed velocity and residual velocity (Eqs. 5–6).
    Hard-pair supervision: use observed target views to construct endpoint paths and optimize the hard flow loss (Eq. 7).
    Soft-pair supervision: construct soft endpoints from same-assigned-cluster candidates and optimize the weighted soft flow loss (Eqs. 8–10).
    Generate missing-view latent representations by few-step ODE inference (Eq. 13).
    Fuse latent views and optimize the prototype-space completion loss (Eqs. 14–16).
    Update model parameters with the overall objective (Eq. 17) and update prototypes by EMA (Eq. 19).
  end for
end for
Form completed representations for all samples and run K-means to obtain final clusters.
return Final cluster assignments.

D Theoretical Analysis

This appendix proves Proposition 1 from Section 3.4 and clarifies the geometric effect and limitations of the prototype component. Proposition 2 separately characterizes the temperature limits of soft-pair aggregation; it is not a prerequisite for the velocity-decomposition result.

D.1 Velocity Decomposition: Proof and Interpretation

Proof of Proposition 1. We analyze velocity regression with the encoder outputs, conditioning inputs, prototype bank, and supervision weights held fixed with respect to the velocity parameters. To express Eq. 11 as a single expectation when both pair sets are nonempty, choose fixed probabilities $p_{\mathrm{h}}, p_{\mathrm{s}} > 0$ with $p_{\mathrm{h}} + p_{\mathrm{s}} = 1$. Select the pair type with these probabilities and then sample from the corresponding distribution in Eq. 7 or Eq. 10. Define

$$\xi = \begin{cases} 1/p_{\mathrm{h}}, & \text{for a hard pair,} \\ \lambda\rho_i^u/p_{\mathrm{s}}, & \text{for a soft pair.} \end{cases} \tag{26}$$

The resulting expectation $\mathbb{E}[\xi\|f_\theta(z_t,t;\eta)-g_t\|^2]$ equals $\mathcal{L}_{\mathrm{FM}}^{\mathrm{hard}}+\lambda\mathcal{L}_{\mathrm{FM}}^{\mathrm{soft}}$. These probabilities only express the sum as a mixture expectation; they do not introduce additional training hyperparameters.

Given a hard- or soft-pair endpoint z_{end}, sample $\epsilon\sim\mathcal{N}(0,I_{d_z})$ and $t\sim\mathrm{Unif}(0,1)$, and define $z_t=(1-t)\epsilon+tz_{\mathrm{end}}$ and $g_t=z_{\mathrm{end}}-\epsilon$. Using X_t, a_t, and b_t from the proposition, the per-sample regression error satisfies the exact identity

$$\begin{aligned}&\xi\,\|a_tb_t+(1-a_t)f^{\mathrm{r}}(X_t)-g_t\|_2^2\\&\quad=\xi(1-a_t)^2\left\|f^{\mathrm{r}}(X_t)-\frac{g_t-a_tb_t}{1-a_t}\right\|_2^2.\end{aligned}\tag{27}$$

The sampled remaining-velocity target is therefore

$$g_t^{\mathrm{r}}=\frac{g_t-a_tb_t}{1-a_t}.\tag{28}$$

This target need not be deterministic at a fixed X_t: different endpoint–noise pairs can yield the same flow state. Thus, a predictor that sees only X_t must minimize the conditional expected loss, rather than equal every sampled target. Conditioned on X_t, the factor $(1-a_t)^2$ is a positive constant and therefore does not affect the pointwise minimizer. Writing $m_\xi(X_t)=\mathbb{E}[\xi\mid X_t]>0$, weighted least squares gives

$$\begin{aligned}f^{\mathrm{r}\star}(X_t)&=\frac{\mathbb{E}[\xi g_t^{\mathrm{r}}\mid X_t]}{m_\xi(X_t)}\\&=\frac{u_\xi(X_t)-a_tb_t}{1-a_t}.\end{aligned}\tag{29}$$

Substitution into Eq. 6 yields $f^\star(X_t)=u_\xi(X_t)$ almost surely on the states with positive conditional weight, proving the proposition. □

For the hard-pair objective alone, $\xi=1$ and the optimal total field is $\mathbb{E}[g_t\mid X_t]$, consistent with conditional velocity regression (Lipman et al., 2023). For the soft-pair objective alone, $\xi=\rho_i^u$ gives the confidence-weighted conditional mean of the constructed targets. The combined loss uses Eq. 26. States with zero conditional weight impose no regression constraint. The factor $(1-a_t)^2$ in Eq. 27 must be retained: dropping it changes the optimization problem for a finite shared network. The result establishes compatibility with the chosen supervision, not improved population expressivity, correct pseudo-cluster assignments, or recovery of an unobserved ground-truth endpoint.

Geometric interpretation and its scope. Hold the trained parameters, condition η, and assigned prototype $c=\mathrm{sg}(c_\eta)$ fixed. Let $\gamma(t)$ denote a generated trajectory satisfying $\dot\gamma(t)=f_\theta(\gamma(t),t;\eta)$, as in Eq. 13, rather than the training interpolation z_t. Define $d_t=\gamma(t)-c$ and let $r_t=f_\theta^{\mathrm{r}}(\gamma(t),t;\eta)$ be the residual network output, whose contribution to the velocity is $(1-a_t)r_t$. For $D_t=\frac{1}{2}\|d_t\|^2$, the complete velocity gives

$$\frac{\mathrm{d}D_t}{\mathrm{d}t}=-a_t\|d_t\|^2+(1-a_t)\langle d_t,r_t\rangle.\tag{30}$$

The prototype component alone contributes a nonpositive term to this distance derivative. At a state with $d_t\neq0$, $\mathrm{d}D_t/\mathrm{d}t<0$ exactly when the weighted residual contribution $(1-a_t)\langle d_t,r_t\rangle$ is smaller than $a_t\|d_t\|^2$. For example, if $\langle d_t,r_t\rangle\le L_t\|d_t\|^2$ and $a_t>(1-a_t)L_t$, then D_t decreases whenever $d_t\neq0$. PRFM does not enforce this bound on the learned residual. Hence this analysis does not guarantee attraction of the full flow, convergence to a true cluster, or monotone distance reduction under a finite Euler step. Online prototype updates occur between training iterations, not within the fixed-prototype trajectory considered here.

D.2 Temperature Effect in Prototype-Prioritized Soft Pairing

Proposition 2 (Temperature limits of soft-pair aggregation) *Fix a nonempty selected candidate set and its latent representations and prototype-similarity scores. As $\tau_p\to0^+$, the soft endpoint converges to the uniform average over candidates attaining the maximum score. As $\tau_p\to\infty$, it converges to the uniform average over the entire selected candidate set.*

Proof. Let $J = \mathcal{A}^u_{i,R}$ denote the selected candidate set and write $s_j = s^u_{ij}$ for $j \in J$. The aggregation weight of candidate j is

$$\alpha_j(\tau_p) = \frac{\exp(s_j/\tau_p)}{\sum_{\ell \in J} \exp(s_\ell/\tau_p)}. \tag{31}$$

Let

$$s_{\max} = \max_{j \in J} s_j, \qquad J_{\max} = \{j \in J \mid s_j = s_{\max}\} \tag{32}$$

denote the maximum score and the set of maximizers, respectively. Subtracting $s_{\max}$ from all logits yields

$$\alpha_j(\tau_p) = \frac{\exp((s_j - s_{\max})/\tau_p)}{\sum_{\ell \in J} \exp((s_\ell - s_{\max})/\tau_p)}. \tag{33}$$

As $\tau_p \to 0^+$, the exponential term converges to zero for every $j \notin J_{\max}$ and remains one for every $j \in J_{\max}$. Hence,

$$\lim_{\tau_p \to 0^+} \alpha_j(\tau_p) = \begin{cases} \dfrac{1}{|J_{\max}|}, & j \in J_{\max}, \\ 0, & j \notin J_{\max}. \end{cases} \tag{34}$$

Because stop-gradient does not change the forward value of the soft endpoint, we obtain

$$\lim_{\tau_p \to 0^+} \widetilde{z}^u_i = \mathrm{sg}\left(\frac{1}{|J_{\max}|} \sum_{j \in J_{\max}} z^u_j \right). \tag{35}$$

In particular, when the maximum score is unique, the soft endpoint converges to the selected candidate most similar to the assigned prototype.

In the opposite limit, $\tau_p \to \infty$, we have $\exp(s_j/\tau_p) \to 1$ for every $j \in J$. Therefore,

$$\lim_{\tau_p \to \infty} \alpha_j(\tau_p) = \frac{1}{|J|}, \tag{36}$$

and

$$\lim_{\tau_p \to \infty} \widetilde{z}^u_i = \mathrm{sg}\left(\frac{1}{|J|} \sum_{j \in J} z^u_j \right). \tag{37}$$

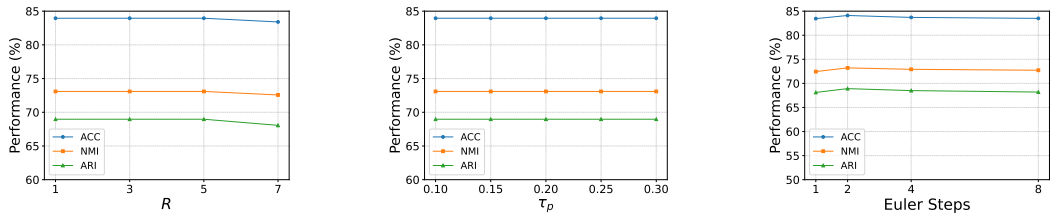
Thus, τ_p controls the aggregation weights over a fixed candidate set; it does not change the preceding top-R selection. These limits characterize the weighting rule, not the correctness of the soft endpoint or its cluster assignment. □

E Hyperparameter Sensitivity Analysis (Q5)

We evaluate the sensitivity of PRFM on HandWritten and YouTubeFace10 at a 70% missing rate. We first vary the candidate size R, the prototype-similarity temperature τ_p, and the number of Euler integration steps. We then jointly vary the prototype guidance scale λ_{pg} and the soft flow-matching loss weight λ, while keeping the remaining settings fixed.

Figure 4 shows that PRFM is generally stable to the candidate size R and the prototype-similarity temperature τ_p on HandWritten, where ACC, NMI, and ARI change only mildly across the tested values. YouTubeFace10 is more sensitive to R: smaller candidate pools give stronger results, while larger pools can degrade ARI, suggesting that overly broad soft targets may include less reliable candidates. For ODE inference, a few Euler steps are sufficient. Four steps give strong performance on YouTubeFace10, whereas further increasing the number of steps does not consistently improve clustering.

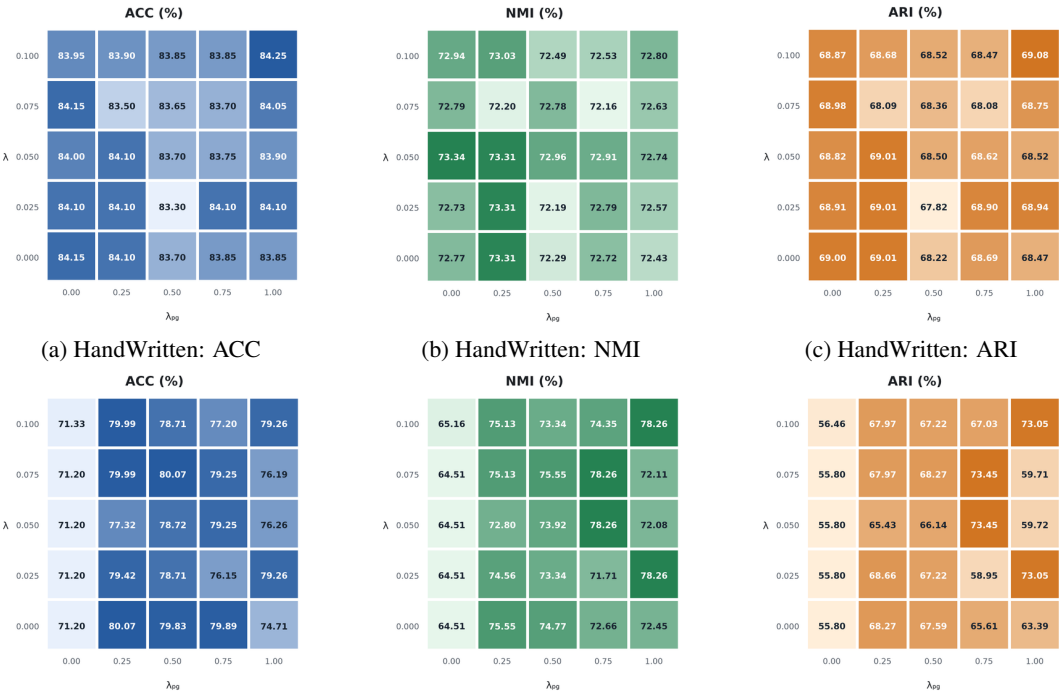
Figure 5 further studies the prototype guidance scale λ_{pg} and the soft flow-matching loss weight λ. HandWritten remains stable over a broad range of settings. On YouTubeFace10, setting $\lambda_{\mathrm{pg}} = 0$ clearly reduces all metrics, supporting the contribution of prototype-routed velocity. Nonzero prototype guidance forms several competitive regions, while the effect of λ is less monotonic. The default setting $\lambda_{\mathrm{pg}} = 0.5$ and $\lambda = 0.05$ provides a balanced choice across datasets and metrics. The boundary value $\lambda_{\mathrm{pg}} = 1$ is included only as a sensitivity stress test; all default and main experiments use $\lambda_{\mathrm{pg}} < 1$, consistent with the remaining-velocity analysis.



(a) HandWritten: R (b) HandWritten: τ_p (c) HandWritten: steps

(d) YouTubeFace10: R (e) YouTubeFace10: τ_p (f) YouTubeFace10: steps

Figure 4: Hyperparameter sensitivity of PRFM to the candidate size R, the prototype-similarity temperature τ_p, and the number of Euler integration steps at a 70% missing rate. The top row shows HandWritten, and the bottom row shows YouTubeFace10.



(a) HandWritten: ACC (b) HandWritten: NMI (c) HandWritten: ARI

(d) YouTubeFace10: ACC (e) YouTubeFace10: NMI (f) YouTubeFace10: ARI

Figure 5: Annotated heatmaps for the sensitivity to the prototype guidance scale λ_{pg} and the soft flow-matching loss weight λ at a 70% missing rate. The top row shows HandWritten, and the bottom row shows YouTubeFace10.

F Convergence Analysis (Q6)

Figure 6 shows the training dynamics of PRFM on HandWritten at a 70% missing rate. ACC, NMI, and ARI increase rapidly in the early epochs and then gradually reach a plateau, indicating that the clustering quality becomes stable after sufficient training. The training loss also shows an overall

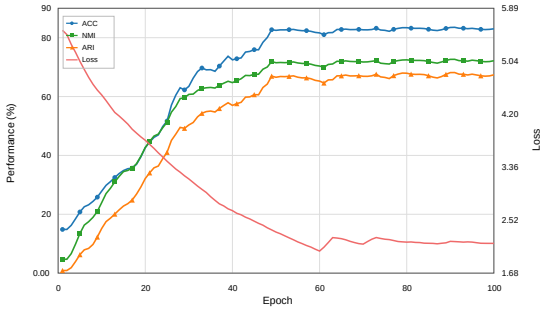


Figure 6: Convergence analysis of PRFM on HandWritten at a 70% missing rate.

decreasing trend, with only minor fluctuations in later epochs. These results suggest that PRFM can be optimized stably and exhibits a clear convergence trend under a high missing rate.

G Runtime Comparison (Q7)

Figure 7 further compares the runtime of representative completion methods: sample-wise point completion (DCP), prototype-based completion (ProImp), diffusion-based completion (DCG and NCD), and PRFM. We report the average latency over HandWritten and YouTubeFace10 at a 70% missing rate, using the first two views and the same batch size as in Table 3. PRFM has competitive train-step latency and much lower inference latency than diffusion-based completion methods, reflecting the efficiency of few-step ODE transport.

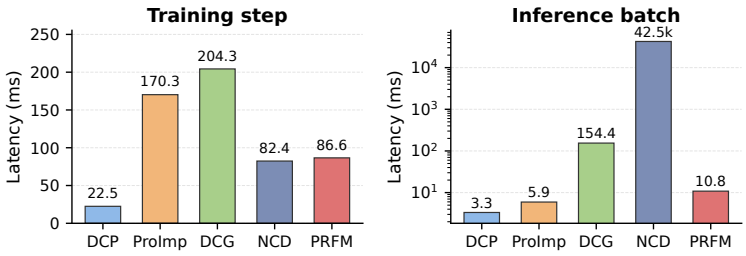


Figure 7: Runtime comparison across representative missing-view completion methods. Bars report average latency on HandWritten and YouTubeFace10 at a 70% missing rate. The inference axis uses a logarithmic scale because NCD requires 1000-step denoising.

AI Use Statement

AI-based tools were used during the preparation of this manuscript solely for language refinement and presentation. Specifically, they assisted with sentence restructuring, grammar and wording improvement, and enhancing the clarity, readability, and overall flow of the text. All AI-assisted content was carefully reviewed and revised by the authors before inclusion in the final manuscript. The authors take full responsibility for all content and conclusions presented in this work.